

## Improved Trajectory Piecewise Linear Combined with POD Model Order Reduction for Subsurface Flow Simulation

Jing Cao<sup>1</sup>, Guo-Jing Chen\*<sup>1</sup>, Jin-Ming Cao<sup>1</sup>, Ya-Xin Fang<sup>1</sup>, Jin-Xiong<sup>1</sup>  
<sup>1</sup>School of Information and Mathematics, Yangtze University, Jingzhou, 434023, China

---

**Abstract:** Reservoir simulation is an indispensable tool for many computational applications such as uncertainty quantification, production optimization and history matching. However, these applications can become computational intractable because the simulation of even a single model is time consuming. To relieve the computational burden, a model order reduction (MOR) procedure referred to as POD-ITPWL is presented. In this procedure, trajectory piecewise linear (TPWL) is improved from linearization point selection and the weight function, and then combined with the POD method. We apply this MOR to a fully implicit oil–water subsurface flow problem. The result demonstrates that the method provides higher accuracy than existing POD-TPWL method, and the speed of POD-ITPWL model is almost the same as POD-TPWL. It provides runtime speedup of about 5 for the full-order model considered in this work.

**Keywords:** reservoir simulation; model order reduce; trajectory piecewise linear; proper orthogonal decomposition

---

Date of Submission: 16-04-2020

Date of Acceptance: 01-05-2020

---

### I. INTRODUCTION

Reservoir simulation is an essential tool for reservoir management such as uncertainty quantification and reservoir design optimization. These applications usually require many simulations. It can pose computational challenges to apply in practice when a large number of models are used. Therefore, how to relieve the computational burden is an urgent problem we need to solve.

Model order reduction (MOR) techniques have shown promise in alleviating computational demands with minimal loss of accuracy. The main goal of MOR method is to generate a reduced-order model of the system, while preserving accurately the input/output behavior of the original system. Consequently, many strategies are based on the concept of projecting the states of the original system onto a suitably selected reduced-order state space to reduce the computational complexity. Trajectory piecewise linearization (TPWL) is a very popular MOR method for nonlinear system, which was originally developed by Rewienski and White [1]. The TPWL method approximates the nonlinear term by a piecewise-linear function obtained by linearizing the system at selected points along its trajectory. It has been applied in several areas, including modeling electronic circuits [2-3], biomedical micro-electromechanical [4], and computational fluid dynamics [5]. TPWL combined with proper orthogonal decomposition (POD) has recently suggested for nonlinear reservoir simulation [6-9]. In POD-TPWL, at each time step, the nonlinear term is linearized around a particular training run solution by the use of the training-run Jacobian and other matrices. The POD basis matrix, which is constructed using the solution states (snapshots) of the training runs output, projects the linearized system to a low-dimensional space. MOR based on POD-TPWL has been shown to provide runtime speedups of  $O(10^2-10^3)$  [6,7,10]. These large speedups result from the fact that the online computations entail only the linear solution of low-order systems, in contrast to the full-order nonlinear solution required for the high-fidelity simulation.

Although the TPWL MOR method in [1] is more efficient for nonlinear systems, it still has some drawbacks. Sometimes, the error is large when the current state is away from most of the expansion points [1,6,11-13]. In this work we improve the TPWL method from two aspects: linearization point selection and the weight function which play key role in TPWL, and then combined with the POD method. The method is referred to as POD-ITPWL. We apply this MOR to oil–water subsurface flow problem to improve the speed and precision of reservoir simulation.

This paper proceeds as follows. In Section 2, we present the fully implicit oil-water subsurface flow governing equations. In Section 3, POD-ITPWL reduced-order modeling is presented. We first provide a detailed description of improved trajectory piecewise linearization (ITPWL), and then generation process of POD basis matrix is described. Then, the derived POD-ITPWL reduced-order modeling is tested in comparison with the exiting POD-TPWL through a two-dimensional oil-water two phase anisotropic reservoir model. Finally, we present conclusion and future work.

## II. GOVERNING EQUATIONS

In this section, we mainly introduce the governing equations for a two-phase (oil–water) system. For a two-phase isothermal immiscible oil–water system, the governing equations are obtained by combining conservation of mass with Darcy’s law for each phase [14]:

$$\nabla \cdot (\alpha \rho_w \vec{v}_w) + \alpha \frac{\partial (\phi S_w \rho_w)}{\partial t} - \alpha \rho_w q_w''' = 0 \quad (1)$$

$$\nabla \cdot (\alpha \rho_o \vec{v}_o) + \alpha \frac{\partial (\phi S_o \rho_o)}{\partial t} - \alpha \rho_o q_o''' = 0 \quad (2)$$

$$\vec{v}_w = -\frac{k_{rw}}{\mu_w} \vec{K} (\nabla p_w - \rho_w g \nabla d) \quad (3)$$

$$\vec{v}_o = -\frac{k_{ro}}{\mu_o} \vec{K} (\nabla p_o - \rho_o g \nabla d) \quad (4)$$

Where subscripts  $w$  and  $o$  are used to identify water and oil,  $S$  is the saturation,  $\rho$  is phase density,  $\phi$  is porosity,  $q'''$  is a source term expressed as flow rate per unit volume,  $t$  is time,  $\alpha$  is a geometric factor,  $\vec{v}$  is the Darcy velocity vector,  $\nabla \cdot$  is the divergence operator,  $k_{rw}$  and  $k_{ro}$  are the relative permeabilities,  $\mu$  is the fluid viscosity,  $\vec{K}$  is the permeability tensor,  $g$  is the acceleration of gravity and  $d$  is depth,  $p$  is pressure,  $\nabla$  is the gradient operator.

Equations (1) and (2) together contain four unknowns,  $p_o$ ,  $p_w$ ,  $S_o$  and  $S_w$ , two of which can be eliminated with aid of the relationships

$$S_o + S_w = 1 \quad (5)$$

$$p_o - p_w = p_c(S_w) \quad (6)$$

Where  $p_c(S_w)$  is the oil–water capillary pressure.

We consider the relatively simple cases and neglect gravitational and capillary pressure effects. Format to discrete in space by using five point block centered finite difference, we may have the nonlinear first-order differential equation (7), see the specific derivation of literature [9]:

$$\underbrace{\begin{bmatrix} \mathbf{V}_{wp} & \mathbf{V}_{ws} \\ \mathbf{V}_{op} & \mathbf{V}_{os} \end{bmatrix}}_{\mathbf{V}} \underbrace{\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{s}} \end{bmatrix}}_{\dot{\mathbf{p}} \text{ and } \dot{\mathbf{s}}} + \underbrace{\begin{bmatrix} \mathbf{T}_w & \mathbf{0} \\ \mathbf{T}_o & \mathbf{0} \end{bmatrix}}_{\mathbf{T}} \underbrace{\begin{bmatrix} \mathbf{p} \\ \mathbf{s} \end{bmatrix}}_{\mathbf{p} \text{ and } \mathbf{s}} = \underbrace{\begin{bmatrix} \mathbf{F}_w(\mathbf{s}) \\ \mathbf{F}_o(\mathbf{s}) \end{bmatrix}}_{\mathbf{F}} \mathbf{q}_{well,t} \quad (7)$$

Where vector  $\mathbf{p}$  and  $\mathbf{s}$  is grid center oil pressure  $p_o$  and water saturation  $S_w$  respectively;  $\dot{\mathbf{p}}$  and  $\dot{\mathbf{s}}$  is the time  $t$  derivative of vector  $\mathbf{p}$  and  $\mathbf{s}$  respectively;  $\mathbf{V}$  is the cumulative matrix;  $\mathbf{T}$  is transmission matrix;  $\mathbf{F}$  is divided flow matrix; Vector  $\mathbf{q}_{well,t}$  is the total flow of oil–water well.

In practice, the source terms are often not the flow rates in the wells but rather the pressures. This can be accounted for by rewriting equation (7) in partitioned form as

$$\begin{bmatrix} \mathbf{V}_{wp,11} & \mathbf{0} & \mathbf{0} & \mathbf{V}_{ws,11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{wp,22} & \mathbf{0} & \mathbf{0} & \mathbf{V}_{ws,22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{V}_{wp,33} & \mathbf{0} & \mathbf{0} & \mathbf{V}_{ws,33} \\ \mathbf{V}_{op,11} & \mathbf{0} & \mathbf{0} & \mathbf{V}_{os,11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{op,22} & \mathbf{0} & \mathbf{0} & \mathbf{V}_{os,22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{V}_{op,33} & \mathbf{0} & \mathbf{0} & \mathbf{V}_{os,33} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{p}}_1 \\ \dot{\mathbf{p}}_2 \\ \dot{\mathbf{p}}_3 \\ \dot{\mathbf{s}}_1 \\ \dot{\mathbf{s}}_2 \\ \dot{\mathbf{s}}_3 \end{bmatrix} + \begin{bmatrix} \mathbf{T}_{w,11} & \mathbf{T}_{w,12} & \mathbf{T}_{w,13} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_{w,21} & \mathbf{T}_{w,22} & \mathbf{T}_{w,23} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_{w,31} & \mathbf{T}_{w,32} & \mathbf{T}_{w,33} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_{o,11} & \mathbf{T}_{o,12} & \mathbf{T}_{o,13} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_{o,21} & \mathbf{T}_{o,22} & \mathbf{T}_{o,23} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_{o,31} & \mathbf{T}_{o,32} & \mathbf{T}_{o,33} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \\ \mathbf{s}_1 \\ \mathbf{s}_2 \\ \mathbf{s}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{w,22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{w,33} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{o,22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{o,33} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \tilde{\mathbf{q}}_{well,t} \\ \mathbf{J}_3(\tilde{\mathbf{p}}_{well} - \mathbf{p}_3) \end{bmatrix} \quad (8)$$

Here, the elements of vector  $\mathbf{p}_1$  are the pressures in those grid blocks that are not penetrated by a well. The elements of  $\mathbf{p}_2$  are the pressures in the blocks where the source terms are prescribed total well flow rates  $\tilde{\mathbf{q}}_{well,t}$ , and those of  $\mathbf{p}_3$  are the pressures in the blocks where the source terms are obtained through prescription of the bottom hole pressures (BHPs)  $\tilde{\mathbf{p}}_{well}$  with the aid of a diagonal matrix of well indices  $\mathbf{J}_3$ . To compute the oil and water flow rates in the wells with prescribed pressures we use the well model

$$\begin{bmatrix} \tilde{\mathbf{q}}_{well,w} \\ \tilde{\mathbf{q}}_{well,o} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{w,33} \\ \mathbf{F}_{o,33} \end{bmatrix} \mathbf{J}_3(\tilde{\mathbf{p}}_{well} - \mathbf{p}_3) \quad (9)$$

To compute the BHPs  $\tilde{\mathbf{p}}_{well}$  in the wells with prescribed total flow rates we need an additional diagonal matrix  $\mathbf{J}_2$  of well indices such that

$$\tilde{\mathbf{q}}_{well,t} = \mathbf{J}_2(\tilde{\mathbf{p}}_{well} - \mathbf{p}_2) \quad (10)$$

From which we obtain

$$\tilde{\mathbf{p}}_{well} = \mathbf{J}_2^{-1} \tilde{\mathbf{q}}_{well,t} + \mathbf{p}_2 \quad (11)$$

To bring these equations in state-space form we define the state vector  $\mathbf{x}$ , input vector  $\mathbf{u}$  and output vector  $\mathbf{y}$  as

$$\mathbf{u} = \begin{bmatrix} \tilde{\mathbf{q}}_{well,t} \\ \tilde{\mathbf{p}}_{well} \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \mathbf{s} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \\ \mathbf{s}_1 \\ \mathbf{s}_2 \\ \mathbf{s}_3 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} \tilde{\mathbf{p}}_{well} \\ \tilde{\mathbf{q}}_{well,w} \\ \tilde{\mathbf{q}}_{well,o} \end{bmatrix} \quad (12,13,14)$$

Equations (8), (9) and (11) can then be rewritten in nonlinear state-space form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{A}(\mathbf{x})\mathbf{x} + \mathbf{B}(\mathbf{x})\mathbf{u} \quad (15)$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}) = \mathbf{C}(\mathbf{x})\mathbf{x} + \mathbf{D}(\mathbf{x})\mathbf{u} \quad (16)$$

where the matrices are defined as

$$\mathbf{A} \square -\mathbf{V}^{-1} \begin{bmatrix} \mathbf{T}_{w,11} & \mathbf{T}_{w,12} & \mathbf{T}_{w,13} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_{w,21} & \mathbf{T}_{w,22} & \mathbf{T}_{w,23} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_{w,31} & \mathbf{T}_{w,32} & \mathbf{T}_{w,33} + \mathbf{F}_{w,33}\mathbf{J}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_{o,11} & \mathbf{T}_{o,12} & \mathbf{T}_{o,13} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_{o,21} & \mathbf{T}_{o,22} & \mathbf{T}_{o,23} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_{o,31} & \mathbf{T}_{o,32} & \mathbf{T}_{o,33} + \mathbf{F}_{o,33}\mathbf{J}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\mathbf{B} \square -\mathbf{V}^{-1} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{w,22} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{w,33}\mathbf{J}_3 \\ \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{o,22} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{o,33}\mathbf{J}_3 \end{bmatrix}, \quad \mathbf{C} \square \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{F}_{w,33}\mathbf{J}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{F}_{o,33}\mathbf{J}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{D} \square \begin{bmatrix} \mathbf{J}_2^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{w,33}\mathbf{J}_3 \\ \mathbf{0} & \mathbf{F}_{o,33}\mathbf{J}_3 \end{bmatrix}$$

It should be noted that the parameters of the system matrix  $\mathbf{A}$  and the input matrix  $\mathbf{B}$  are still functions of the state  $\mathbf{x}$ . Vector  $\mathbf{y}$  is called the output vector.  $\mathbf{C}$  is the output matrix.  $\mathbf{D}$  is the direct transfer matrix. Because the elements of the matrix  $\mathbf{V}$ ,  $\mathbf{T}$ ,  $\mathbf{F}$ ,  $\mathbf{J}$  are function of the state variables  $\mathbf{x}$ , the system is a nonlinear system.

### III. POD-ITPWL REDUCED-ORDER MODELING FOR SUBSURFACE FLOW

In this section, POD-ITPWL reduced-order modeling is presented. We first provide a detailed description of improved trajectory piecewise linearization (ITPWL), and then generation process of POD basis matrix is described.

#### 3.1 Improved Trajectory Piecewise Linearization Method

In order to reduce the computational effort associate with full-order solution of Eq. (15), we now introduce the improved trajectory piecewise linearization (ITPWL) method. In TPWL [1] methodology a nonlinear system is represented as a weighted combination of piecewise linear systems and each linear system is projected into a low-dimensional space using an appropriate projection method.

For reservoir simulation nonlinear system (15), we first simulate one or more full-order “training” runs using a specified set for each well. We save the states and the derivative information at each time step of one or more of the training runs. We obtain a set of expansion points  $\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_{s-1}$  by using a suitable linearization point selection method, and then linearize the nonlinear term  $\mathbf{f}(\mathbf{x}) = \mathbf{A}(\mathbf{x})\mathbf{x}$  at various expansion points:

$$\dot{\mathbf{x}} = \mathbf{G}_i \mathbf{x} + (\mathbf{f}(\hat{\mathbf{x}}_i) - \mathbf{G}_i \hat{\mathbf{x}}_i) + \mathbf{B}_i \mathbf{u}, \quad i = 0, 1, \dots, (s-1) \quad (17)$$

Where  $\mathbf{G}_i$  is Jacobian matrix of  $\mathbf{f}(\mathbf{x})$  at  $\hat{\mathbf{x}}_i$ ,  $\mathbf{B}_i = \mathbf{B}(\hat{\mathbf{x}}_i)$ .

The nonlinear system (15) is approximated by constructing a weighted combination of linearized models (17):

$$\dot{\mathbf{x}} = \sum_{i=0}^{s-1} \tilde{\omega}_i(\mathbf{x})(\mathbf{G}_i \mathbf{x} + (\mathbf{f}(\hat{\mathbf{x}}_i) - \mathbf{G}_i \hat{\mathbf{x}}_i) + \mathbf{B}_i \mathbf{u}) \quad (18)$$

where  $\tilde{\omega}_i(\mathbf{x})$  are state-dependent weights. (All the weights are nonnegative, and satisfy  $\sum_{i=0}^{s-1} \tilde{\omega}_i(\mathbf{x}) = 1$ , for all  $\mathbf{x}$ ).

The vectors and matrices in Eq. (18) are still  $O(N_c^k)$  ( $N_c$  is the total number of grid blocks), with  $k = 1, 2$ , respectively. To reduce the size of the system, we apply dimension reduction. Assuming we have already generated  $l$ -th order bases  $\Phi_l \in \mathbb{R}^{N_c \times l}$ , and then perform a change of variables  $\mathbf{x} = \Phi_l \mathbf{z}$ , where  $\mathbf{z} \in \mathbb{R}^l$  is a low-dimensional variable. Reduction is achieved because  $l \ll N_c$ . Projecting (18) yields:

$$\dot{\mathbf{z}} = \sum_{i=0}^{s-1} \tilde{\omega}_i(\Phi_l \mathbf{z})(\mathbf{G}_{il} \mathbf{z} + \Phi_l^T (f(\hat{\mathbf{x}}_i) - \mathbf{G}_i \hat{\mathbf{x}}_i) + \mathbf{B}_{il} \mathbf{u}) \quad (19)$$

Where  $\mathbf{z} = \Phi_l^T \mathbf{x}$ ,  $\mathbf{G}_{il} = \Phi_l^T \mathbf{G}_i \Phi_l$ ,  $\mathbf{B}_{il} = \Phi_l^T \mathbf{B}_i$ .

In fact, since computing  $\tilde{\omega}_i(\Phi_l \mathbf{z})$  is inexpensive, instead of weights  $\tilde{\omega}_i(\Phi_l \mathbf{z})$ , we may use weighting functions  $\omega_i(\mathbf{z})$  which depend on the reduced order state vector. Replacing  $\tilde{\omega}_i(\Phi_l \mathbf{z})$  with  $\omega_i(\mathbf{z})$  in (19) yields the following reduced order model:

$$\begin{cases} \dot{\mathbf{z}} = \sum_{i=0}^{s-1} \omega_i(\mathbf{z})(\mathbf{G}_{il} \mathbf{z} + \Phi_l^T (f(\hat{\mathbf{x}}_i) - \mathbf{G}_i \hat{\mathbf{x}}_i) + \mathbf{B}_{il} \mathbf{u}) \\ \mathbf{y} = \mathbf{C}_l \mathbf{z} + \mathbf{D} \mathbf{u} \end{cases} \quad (20)$$

where  $\mathbf{C}_l = \mathbf{C} \Phi_l$ .

TPWL shows some problems during simulation. The error is large when the current state is away from most of the expansion points. We will improve the TPWL method from two aspects: linearization point selection and the weight function which play key role in TPWL.

In order to obtain high quality linear expansion, we use a new linearization point selection method based on a global maximum error controller for TPWL model order reduction [3]. This method is based on a simple fact that the simulation cost of the TPWL model is very low, and selects the state at which the responses of the current TPWL model and the full nonlinear model have the maximum difference as a new linearization point. It can generate the TPWL model of smaller size and higher accuracy. In reservoir simulation, the specific process of the algorithm is described as:

- 1) Give the maximum error control limit  $\alpha$  and input vector  $\mathbf{u}(t)$ ;
- 2) Simulate the full-order reservoir simulator and save the output state vectors  $\{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_M\}$ ;
- 3) The initial state  $\mathbf{x}_0$  is taken as the first linear expansion point  $\hat{\mathbf{x}}_0$ , and set  $i = 1$ ;
- 4) Use the TPWL method to establish a temporary model:

$$\dot{\mathbf{z}} = \sum_{j=0}^{i-1} \tilde{\omega}_j(\Phi_l \mathbf{z})(\mathbf{G}_{jl} \mathbf{z} + \Phi_l^T (f(\hat{\mathbf{x}}_j) - \mathbf{G}_j \hat{\mathbf{x}}_j) + \mathbf{B}_{jl} \mathbf{u}) \quad (21)$$

- 5) The model (21) is simulated and the state vector  $\{\tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_M\}$  is obtained;
  - 6)  $\{\tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_M\}$  and  $\{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_M\}$  will be compared to find the maximum error state  $\tilde{\mathbf{x}}_k$ , and then record the maximum error  $\eta_{\max}$  and  $k$ ;
  - 7) If  $\eta_{\max} > \alpha$ , select the first  $i + 1$  linearization point  $\hat{\mathbf{x}}_{i+1} = \tilde{\mathbf{x}}_k$ , and set  $i = i + 1$ , then turn to 4);
- If  $\eta_{\max} < \alpha$ , the loop ends and the linearization point  $\{\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_{i-1}\}$  is returned.

By experiments, we use the following weight function that is more effective [2]:

$$\omega_i(\mathbf{z}) = \left[ \frac{d_{\min}}{d_i(\mathbf{z})} e^{-\frac{d_i(\mathbf{z}) - d_{\min}}{D_{\min}}} \right]^p \quad (22)$$

Where  $d_i(\mathbf{z}) = \|\mathbf{z} - \hat{\mathbf{z}}_i\|_2^2$ ,  $d_{\min} = \min(d_i(\mathbf{z}))$ ,  $i = 0, 1, \dots, (s-1)$ .  $D_{\min}$  is the minimum distance among those center points  $\{\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_{s-1}\}$ . Parameter  $p$  is between 1 and 2. The whole weight function is finally

normalized to satisfy  $\sum_{i=0}^{s-1} \omega_i(\mathbf{z}) = 1$ .

### 3.2 Proper Orthogonal Decomposition (POD) Method

In this work, we construct the  $\Phi_l$  matrix through application of POD. First, training simulations of the full-order model is performed, and then solution snapshots are saved. As discussed in [15-16], we separately treat the pressure and water saturation snapshots, and construct basis matrices for each. The snapshots are assembled into data matrices  $\mathbf{X}_p \in \mathbb{R}^{N_c \times m}$  and  $\mathbf{X}_s \in \mathbb{R}^{N_c \times m}$ :

$$\mathbf{X}_p = [\mathbf{x}_p^1 \ \mathbf{x}_p^2 \ \cdots \ \mathbf{x}_p^m], \quad \mathbf{X}_s = [\mathbf{x}_s^1 \ \mathbf{x}_s^2 \ \cdots \ \mathbf{x}_s^m] \quad (23)$$

Where  $m$  denotes the total number of time steps in all of training simulations.

After the snapshots are obtained, the mean of the snapshots is computed:

$$\bar{\mathbf{x}}_p = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_p^i, \quad \bar{\mathbf{x}}_s = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_s^i$$

The data matrices  $\hat{\mathbf{X}}_p$  and  $\hat{\mathbf{X}}_s$  are determined by subtracting the mean from each snapshot:

$$\hat{\mathbf{X}}_p = [\mathbf{x}_p^1 - \bar{\mathbf{x}}_p, \mathbf{x}_p^2 - \bar{\mathbf{x}}_p, \cdots, \mathbf{x}_p^m - \bar{\mathbf{x}}_p]$$

$$\hat{\mathbf{X}}_s = [\mathbf{x}_s^1 - \bar{\mathbf{x}}_s, \mathbf{x}_s^2 - \bar{\mathbf{x}}_s, \cdots, \mathbf{x}_s^m - \bar{\mathbf{x}}_s]$$

We then perform singular value decomposition (SVD) on  $\hat{\mathbf{X}}_p$  and  $\hat{\mathbf{X}}_s$ , and preserve the first  $l_p$  and  $l_s$  left singular vectors. The values of  $l_p$  and  $l_s$  are determined based on the energy criterion. The resulting two basis matrices are created,  $\Phi_{lp}$  and  $\Phi_{ls}$  for the oil pressure and water saturation states, respectively. We then assemble into the global basis matrix  $\Phi_l$ , where  $l = l_p + l_s$ . The  $\Phi_l$  will project Eq. (18) into a low-dimensional space.

#### IV. NUMERICAL RESULTS

In this section, we present numerical results for POD-TPWL and POD-ITPWL for a two-dimensional oil-water two phase anisotropic reservoir model. It contains  $21 \times 21$  grid blocks, and the permeability field and porosity field is shown in Fig. 1, 2. The physical dimension of each grid-block is  $33 \times 33 \times 2$ m. The viscosity  $\mu_o$  of the crude oil is 5mPa·s, and formation water viscosity  $\mu_w$  is 1mPa·s. The comprehensive compression coefficient  $c_t$  is  $3.0 \times 10^{-3} \text{MPa}^{-1}$ . Original formation pressure  $p_i$  is 30MPa. Borehole radius  $r_{well}$  is 0.114m. The end point relative permeability of oil phase  $k_{ro}^0$  is 0.9, and the end point relative permeability of water phase  $k_{rw}^0$  is 0.6. Oil phase Corey index  $n_o$  is 2.0, and water phase Corey index  $n_w$  is 2.0. Residual oil saturation  $S_{or}$  is 0.2, and irreducible water saturation  $S_{wc}$  is 0.2. There are a injection well and four production wells in a five-spot pattern. Capillary pressure and gravity effects are neglected.

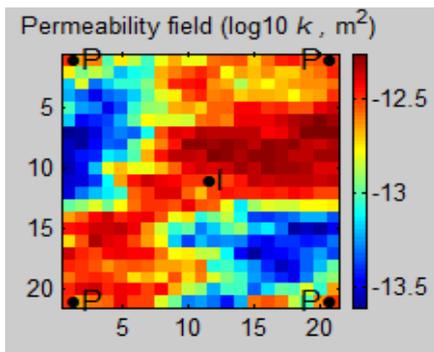


Fig.1 Permeability field

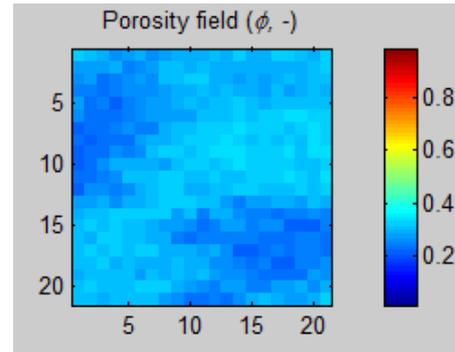


Fig.2 Porosity field

The first step in the POD-TPWL and POD-ITPWL procedures is to perform training runs using the full-order model to generate the states and Jacobian matrices. For this reservoir model a training simulation is performed. In this simulation the injection well is prescribed to water injection rate of  $86.4 \text{m}^3/\text{d}$ . This condition will be maintained in subsequent test runs (our focus here is on varying the production well BHPs). For the production wells, the BHPs are prescribed as 26MPa.

In the training runs, we simulate reservoir performance for a total of 1200 days with a maximum time step of 20 days. From the training runs 62 pressure and saturation snapshots and Jacobian matrices are recorded. The POD basis matrix is constructed following the approach described in section 3.2. The reduced basis  $\Phi_l$

contains a total of 46 columns, 22 of which correspond to pressure states and 24 to saturation states. For POD-TPWL method, we obtain 10 linearization points, and for the POD-ITPWL method we obtain 14 linearization points.

We next consider two different test cases to evaluate the predictive capability of POD-TPWL and POD-ITPWL reduced order model (ROM) . In a test case, the BHP schedules differ from those used in the training run.

(1) Prediction Using ROM-Test I

We change the BHP of the four production wells, and they are set to 23MPa. The difference is little compared to the bottom-hole pressure of training simulation. The injection well water injection rate is the same as in the training simulation.

Results are shown for oil production rates in four wells (Figs.3-6) and water production rates in well P2 and P3 (Figs.7 and 8). We focus on well P2 and P3 for water production rates since the water production rates of well P1 and P4 are close zero. In those figures, the black curves denote full-order model solution. The blue and red curves depict the POD-TPWL and POD-ITPWL solutions, respectively.

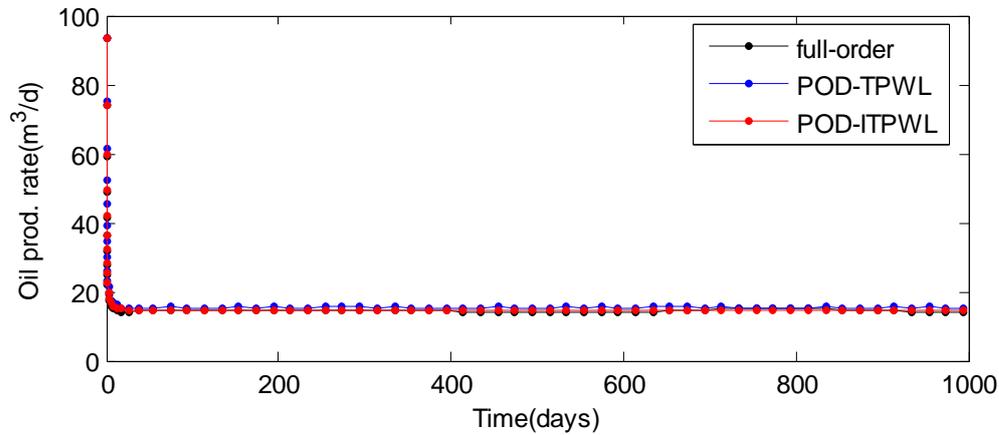


Fig.3 Oil production rate P1 (Test I)

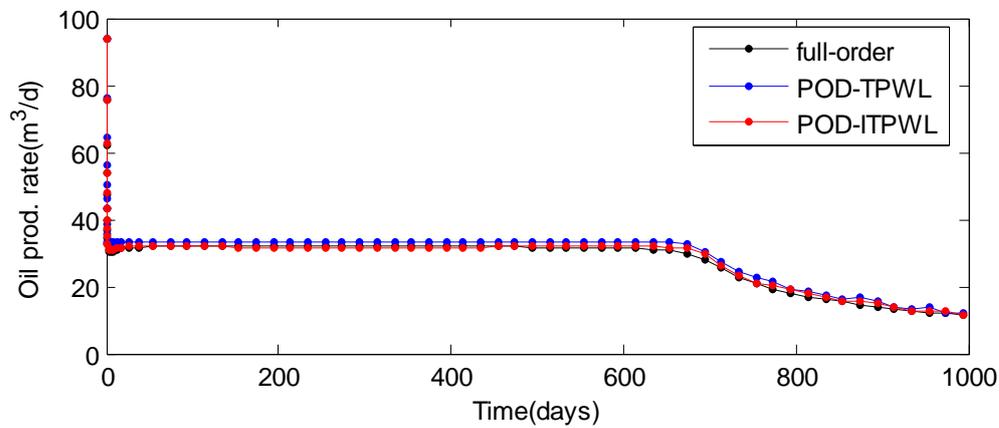


Fig.4 Oil production rate P2 (Test I)

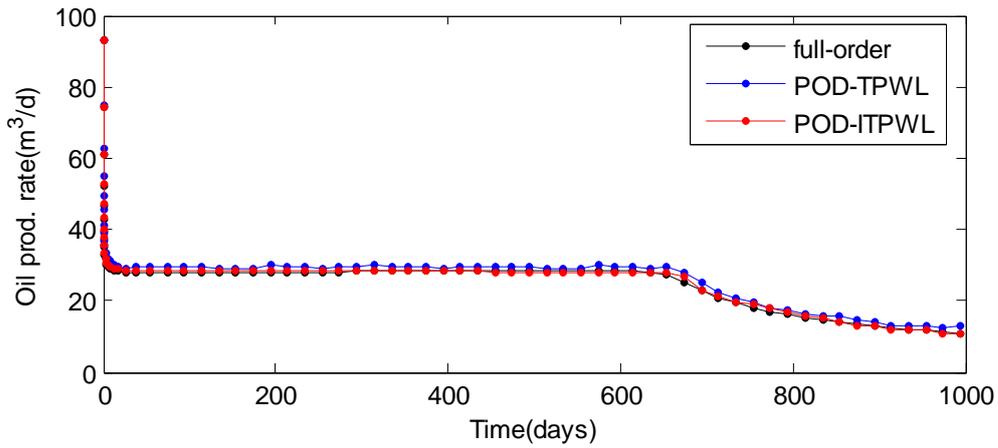


Fig.5 Oil production rate P3 (Test I)

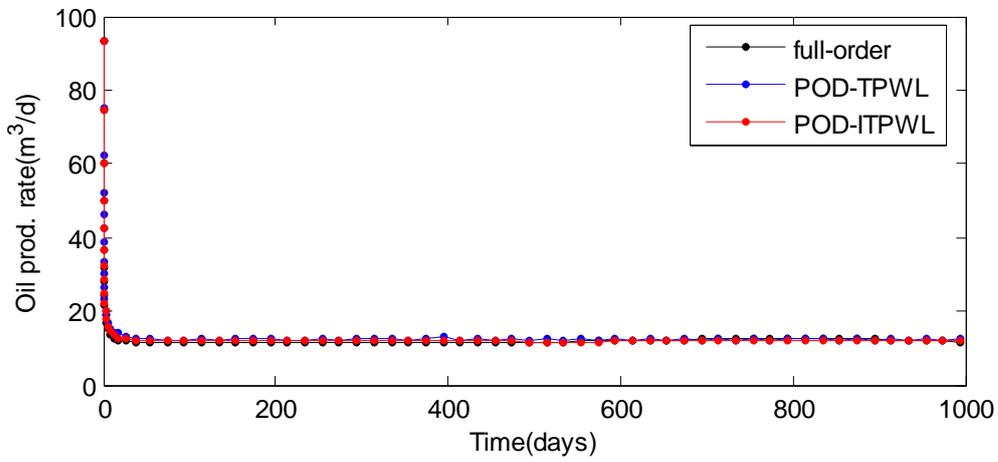


Fig.6 Oil production rate P4 (Test I)

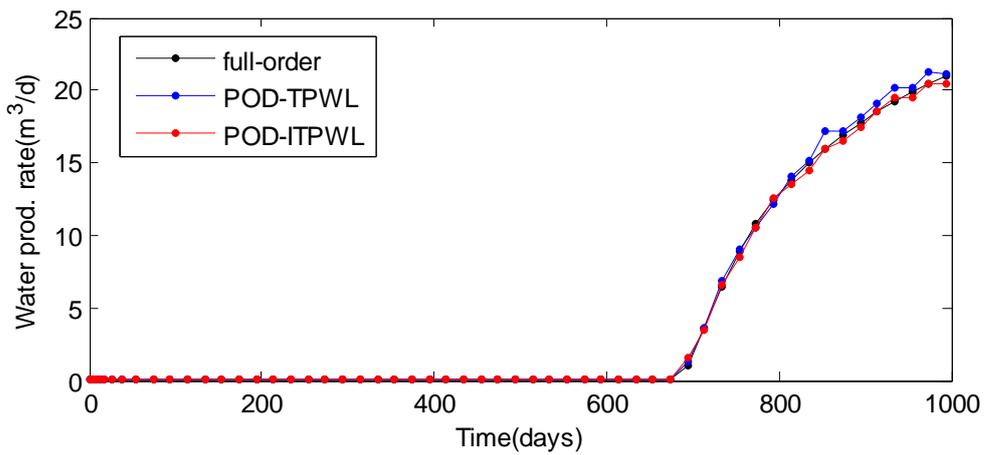
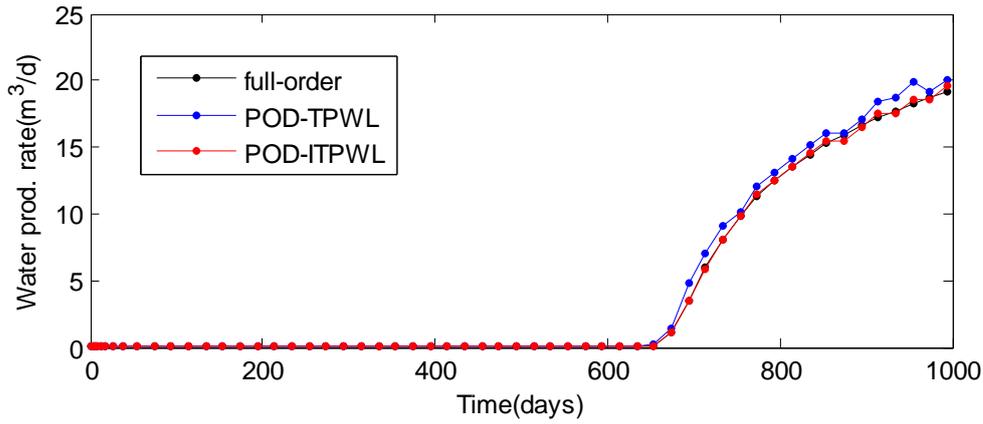


Fig.7 Water production rate P2 (Test I)



**Fig.8** Water production rate P3 (Test I)

In order to assess the quality of the reduced-order models, we now define error which is expressed as percentage. The time-average error  $E_o$  in oil production rate and the time-average error  $E_w$  in water production rate are given as:

$$E_{o,w} = \frac{1}{n_{p_w}} \sum_{j=1}^{n_{p_w}} \frac{\int_0^T |q_{o,w_{FOM}}^j - q_{o,w_{ROM}}^j| dt}{\int_0^T q_{o,w_{FOM}}^j dt} \times 100\% \quad (24)$$

Here  $q_{o,w_{FOM}}^j$  and  $q_{o,w_{ROM}}^j$  are the oil or water production rate for well  $j$  by the full-order and reduced-order (POD-TPWL or POD-ITPWL) model, respectively.  $T$  is the total simulation time,  $n_{p_w}$  is the number of production wells. Median errors for the results of POD-TPWL and POD-ITPWL model are presented in Table 1.

**Table 1** Median errors in POD-TPWL and POD-ITPWL results (Test I)

	POD-TPWL error	POD-ITPWL error
Oil production rate	6.2%	2.1%
Water production rate	7.1%	2.4%

From these figures and table, we see that both ROMs are able to provide useful results when the test-case BHPs are close to those used in training, however, the POD-ITPWL solution is more accurate than POD-TPWL.

The simulation times for the full-order reservoir simulation, the POD-TPWL, and the POD-ITPWL model are shown in Table 2. The POD-ITPWL model reduces the simulation time by about a factor of five.

**Table 2** Comparison of simulation time (Test I)

	full-order	POD-TPWL	POD-ITPWL
Time	94.27s	17.18s	18.82s

## (2) Prediction Using ROM-Test II

For the test II, four production well BHPs are set to 20MPa. The difference is larger compared with the bottom hole pressure of training simulation. The specification for the injection well is the same as in the previous case. Results are shown for oil production rates in four wells (Figs.9-12) and water production rates in well P2 and P3 (Figs.13 and 14). We focus on well P2 and P3 for water production rates since the water production rates of well P1 and P4 are also close zero. In those figures, the black curves denote full-order model solution. The blue and red curves depict the POD-TPWL and POD-ITPWL solutions, respectively.

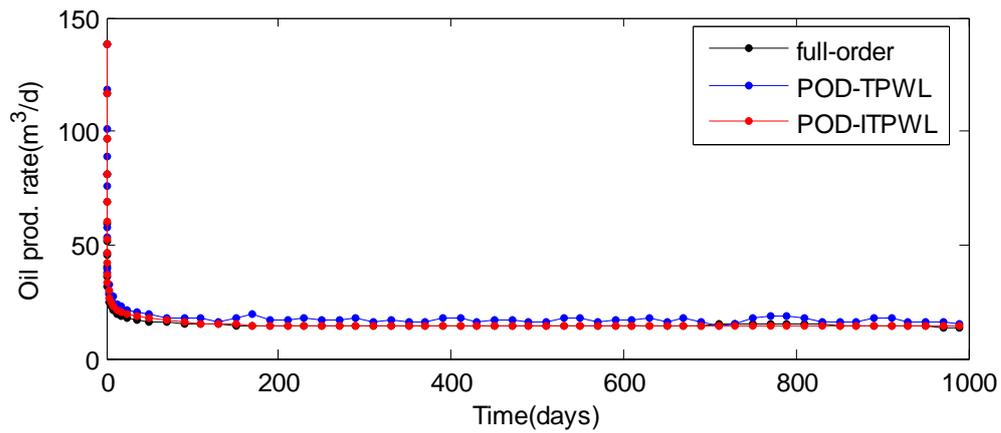


Fig.9 Oil production rate P1 (Test II)

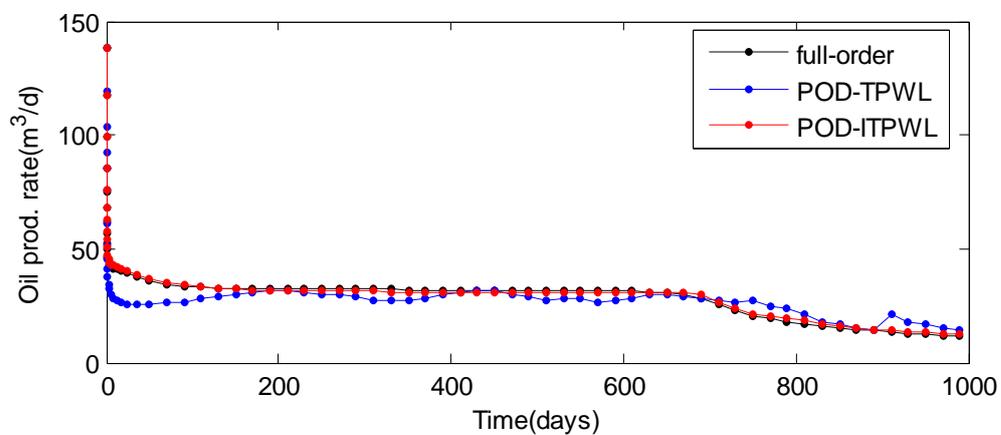


Fig.10 Oil production rate P2 (Test II)

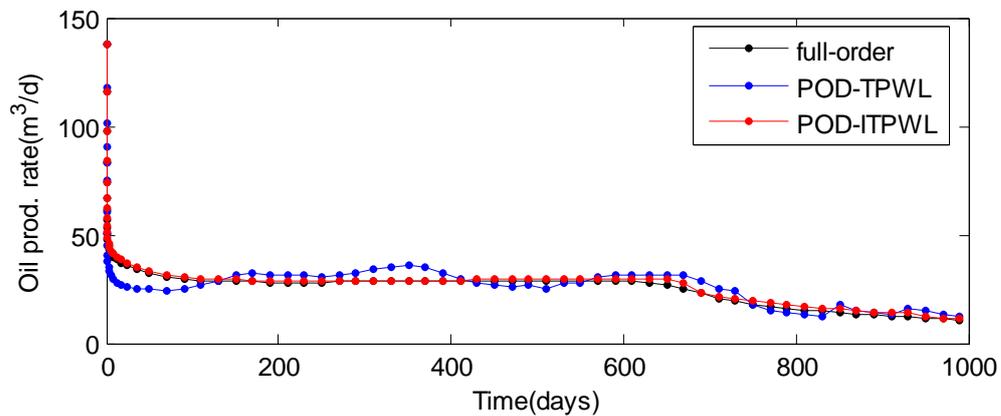


Fig.11 Oil production rate P3 (Test II)

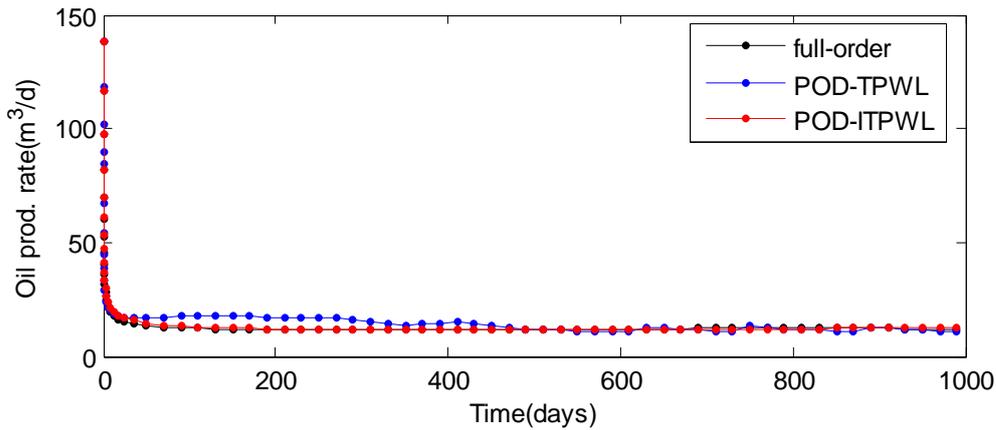


Fig.12 Oil production rate P4 (Test II)

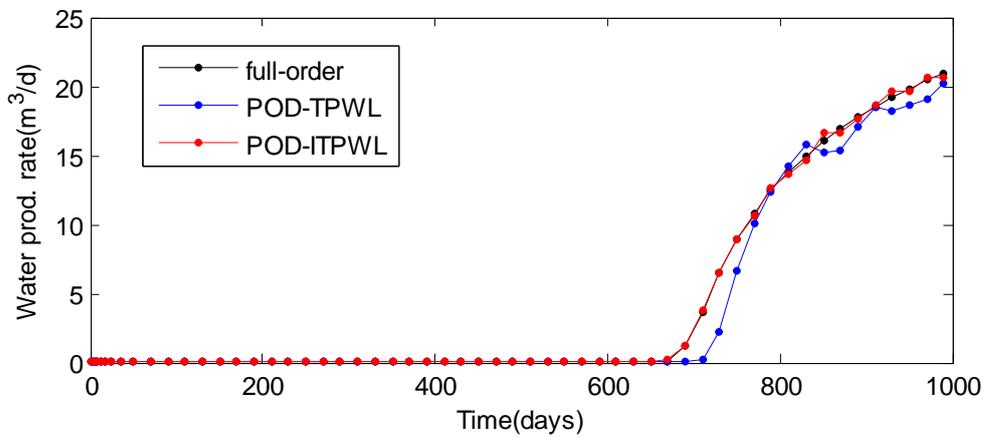


Fig.13 Water production rate P2 (Test II)

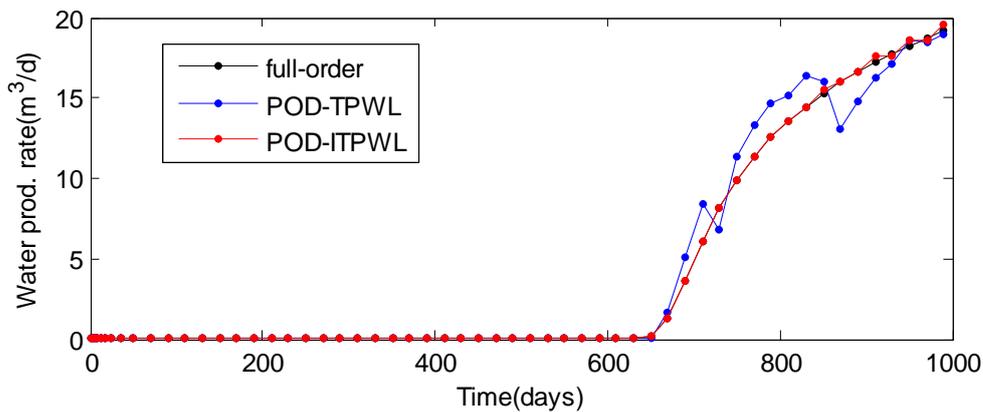


Fig.14 Water production rate P3 (Test II)

Median errors for the results of POD-TPWL and POD-ITPWL model are presented in Table3.

**Table 3** Median errors in POD-TPWL and POD-ITPWL results (Test II)

	POD-TPWL error	POD-ITPWL error
Oil production rate	15.2%	3.0%
Water production rate	16.6%	2.6%

From these figures and table, the results demonstrate that when the difference of production well BHPs is larger compared to the bottom hole pressure of training simulation, the accuracy of POD-TPWL model degrades, however, POD-ITPWL model still leads to solutions in relatively close agreement with the full-order

model.

For test II, the simulation times are given in table 4. The ROM with POD-ITPWL is also able to approximately reduce the simulation time by 5 times compared to time for the full-order reservoir model.

**Table 4** Comparison of simulation time (schedule 2)

	full-order	POD-TPWL	POD-ITPWL
Time	96.21s	18.62s	19.34s

The offline preprocessing cost for ROMs can be relatively high. However, in applications such as optimization that require very large numbers (e.g., thousands) of simulations, this overhead is small relative to the cost of using full-order for all simulation. In such cases, the use of ROMs can provide significant reduction in computational cost.

## V. CONCLUSION

In this work we introduce an improved MOR called POD-ITPWL. This MOR improves TPWL method from linearization point selection and the weight function, and then combined with the POD method. The method thus provides higher accuracy than existing POD-TPWL method. The speed of POD-ITPWL model is almost the same as POD-TPWL model. It provides runtime speedup of about 5 for the quite small full-order model considered in this work (greater speedups should be obtained for the larger model).

In future work we could enable the use of POD-ITPWL for the larger and much more complicated subsurface flow models.

## REFERENCES

- [1]. M. Rewienski, J. White. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 22(2): 155–170, 2003.
- [2]. N. Dong and J. Roychowdhury. General-purpose nonlinear model-order reduction using piecewise-polynomial representations. *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, 27(2): 249-264, 2008.
- [3]. Y. Liu, W. Yuan, H. Chang. A global maximum error controller-based method for linearization point selection in trajectory piecewise-linear model order reduction. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 33(7): 1100–1104, 2014.
- [4]. D. Vasilyev, M. Rewienski, and J. White. Macromodel generation for BioMEMS components using a stabilized balanced truncation plus trajectory piecewiselinear approach. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(2):285–293, 2006.
- [5]. David Gratton. Reduced-order trajectory piecewise-linear models for nonlinear computational fluid dynamics. In: 34th AIAA Fluid Dynamic Conference, 2004.
- [6]. M. A. Cardoso, L. J. Durllofsky. Linearized reduced-order models for subsurface flow simulation. *Journal of Computational Physics*, 229(3): 681-700, 2010.
- [7]. J. He, J. Satrom, L. J. Durllofsky. Enhanced linearized reduced-order models for subsurface flow simulation. *Journal of Computational Physics*, 230(23): 8313-8341, 2011.
- [8]. J. He, L. J. Durllofsky. Reduced-order modeling for compositional simulation by use of trajectory piecewise linearization. *SPE Journal*, 19(5):858-872, 2014.
- [9]. J. He, L. J. Durllofsky. Constraint reduction procedures for reduced-order subsurface flow models based on POD-TPWL. *International Journal for Numerical Methods in Engineer*, 103 (1):1-30, 2015.
- [10]. Sumeet Trehan, Louis J. Durllofsky. Trajectory piecewise quadratic reduced-order model for subsurface flow, with Application to PDE-constrained optimization. *Journal of Computational Physics*, 326: 446-473, 2016.
- [11]. Seonkyoo Yoon, Zeid M. Alghareeb, John R. Williams. Hyper-Reduced-Order Models for Subsurface Flow Simulation. *SPE Journal*, 21(6):2128-2140, 2016.
- [12]. S. A. Nahvi, M. Nabi, and S. Janardhanan. Trajectory Based Methods for Nonlinear MOR: Review and Perspectives. *IEEE Int. Conf.Signal Processing, Computing and Control*, 2012, 1-6.
- [13]. VOSS T., PULCH R., TERMATEN J., ELGUENNOUNI A. Trajectory Piecewise Linear Approach for Nonlinear Differential-Algebraic Equations in Circuit Simulation. *Scientific Computing in Electrical Engineering, Mathematics in Industry*, 11: 167-173, 2007.
- [14]. J. D. Jansen. *Systems Description of Flow Through Porous Media*. Springer: Springer Briefs in Earth Sciences, 2013, 21-36.
- [15]. M. A., Cardoso, L. J., Durllofsky, and P., Sarma. Development and Application of Reduced-Order Modeling Procedures for Subsurface Flow Simulation. *International Journal for Numerical Methods in Engineering*, 77(9): 1322-1350, 2008.
- [16]. J. F., van Doren, R. Markovinovic, J. D., Jansen. Reduced-Order Optimal Control of Water Flooding using Proper Orthogonal Decomposition. *Computational Geosciences*, 10: 137-158, 2006.

Guo-Jing Chen, et al. "Improved Trajectory Piecewise Linear Combined with POD Model Order Reduction for Subsurface Flow Simulation." *International Journal of Engineering Science Invention (IJESI)*, Vol. 09(04), 2020, PP 45-56.